L12: Entry 1 of 30 File: USPT Nov 6, 2001

DOCUMENT-IDENTIFIER: US 6314526 B1
TITLE: Resource group quorum scheme for highly scalable and highly available cluster system management

PCPR:
The present invention is related to the subject matter of commonly assigned, copending U.S. patent applications Ser. No. 09/164,130 (Docket No. AT9-97-760) entitled "A Rule-Based Cluster System Management Model" and filed Sep. 30, 1998 and Ser. No. 09/114,051 (Docket No. AT9-97-761) entitled "A Highly Scalable and Highly Available Cluster System Management Scheme" and filed Jul. 10, 1998. The content of the above-referenced applications are incorporated herein by reference.

BSPR:
A cluster system, also referred to as a cluster multiprocessor system (CMP) or simply as a "cluster," is a set of networked data processing systems with hardware and software shared among those data processing systems, typically but not necessarily configured to provide highly available and highly scalable application services. Cluster systems are frequently implemented to achieve high availability, an alternative to fault tolerance for mission-critical applications such as aircraft control and the like. Fault tolerant data processing systems rely on specialized hardware to detect hardware faults and switch to a redundant hardware component, regardless of whether the component is a processor, memory board, hard disk drive, adapter, power supply, etc. While providing seamless cutover and uninterrupted performance, fault tolerant systems are expensive, due to the redundant hardware requirement, and fail to address software errors, a more common source of data processing system failure.

BSPR:
High availability utilizes standard hardware, but provides software allowing resources to be shared system wide. When a node, component, or application fails, an alternative path to the desired resource is quickly established. The brief interruption required to reestablish availability of the resource is acceptable in many situations. The hardware costs are significantly less than fault tolerant systems, and backup facilities may be utilized during normal operation. An example of the software utilized for these purposes is the HACMP (High Availability Cluster Multiprocessing) for AIX.RTM. (Advanced Interactive Executive) software available from International Business Machines Corporation of Armonk, N.Y. and the RS6000 SP software available from International Business Machines Corporation.

BSPR:
In order to keep a distributed database in a consistent state at all times, a two-phase commit protocol may be utilized. For a fully replicated database (i.e. every data processing system has a copy), 2N messages must be exchanged for each write operation, where N is the number of data processing systems in the cluster. Thus, while the size of a cluster configuration/status database grows linearly with respect to cluster size, access time to the database grows either linearly or logarithmically with respect to cluster size. Moreover, when bringing up a cluster, the number of events (and therefore the amount of status information which needs to be updated) grows linearly with respect to cluster size. Hence, the time or cost required to bring up a cluster with a

fully replicated distributed cluster configuration database grows on the order of N.sup.2. The complexity of cluster system management may thus be characterized as being on the order of N.sup.2. For very large scale cluster systems (over 1,000 data processing systems), full replication of the cluster configuration database becomes unwieldy.

BSPR:
Another critical issue in highly available cluster systems is how to handle network partitions. Network partitions occur if a cluster is divided into two or more parts, where data processing systems in one part cannot communicate with data processing systems in another part. When a network partition occurs, it is crucial not to run multiple copies of the same application, especially a database application such as the cluster configuration database, from these (temporarily) independent parts of the cluster. A standard way of handling this problem is to require that a cluster remain offline unless it reaches quorum. The definition of quorum varies. In some implementations, a majority quorum is employed and a portion of the cluster is said to have reached quorum when the number of active servers in that portion is at least N/2+1. A different scheme may require a smaller number of servers to be active to reach quorum as long as the system can guarantee that at most only one portion of the cluster can reach quorum. In a very large scale cluster, the condition for quorum tends to be too restrictive. A majority quorum is used herein, although the invention is applicable to other forms of quorum.

BSPR:
Thus, when a network partition occurs, only the portion of the cluster (if any) which contains the majority of the data processing systems in the cluster may run applications. Stated differently, no services are provided by the cluster unless at least one half of the data processing systems within the cluster are online.

BSPR:
It would be desirable, therefore, to provide a mechanism for maintaining a distributed database containing cluster configuration information without incurring the costs associated with full replication. It would further be advantageous for the mechanism to be scalable and applicable to clusters of any size, even those larger than 1,000 data processing systems. It would further be advantageous to permit cluster portions to continue providing services after a network partition even if a quorum has not been reached.

DRPR:
FIG. 4 is a high level flowchart for a process of handling node failure within a cluster system including resource groups in accordance with a preferred embodiment of the present invention.

DEPR:
With reference now to the figures, and in particular with reference to FIG. 1, a block diagram of a cluster multi-processing system in which a preferred embodiment of the present invention may be implemented is depicted. System 102 includes a plurality of server nodes 104-110, each typically identified by a unique name. Each node 104-110 may be a symmetric multi-processor (SMP) data processing system such as a RISC System/6000.RTM. system available from International Business Machines Corporation of Armonk, N.Y. or a data processing system functioning as a Windows NT.TM. server.

DEPR:
Each node 104-110 within system 102 includes an operating system, such as the Advanced Interactive Executive (AIX.RTM.) operating system available from International Business Machines Corporation of Armonk, N.Y. or the Windows NT.TM. operating system available from Microsoft Corporation of Redmond, Wash. Nodes 104-110 within system 102 also include high availability cluster software capable of running on top of or in conjunction with the operating system. This high availability cluster software includes the features described below.

DEPR:
Nodes 104-110 are connected to public local area networks 112-114, which may be an Ethernet, Token-Ring, fiber distributed data interface (FDDI), or other network. Public networks 112-114 provide clients 116-120 with access to servers 104-110. Clients 116-120 are data processing systems which may access, each running a "front end" or client application which queries server applications running on nodes 104-110.

DEPR:
Typically, each node 104-110 runs server or "back end" applications which access data on shared external disks 122-126 via disk buses 128-130. Nodes 104-110 may also be connected by an additional network 132 or networks. For example, a private network may provide point-to-point connection between nodes 104-110 within system 102, with no access provided to clients 116-120. The private network, if available, may be utilized for lock traffic, and may be an Ethernet, Token-Ring, FDDI, or serial optical channel connector (SOCC) network. A serial network may also provide point-to-point communication between nodes 104-110, used for control messages and heartbeat traffic in the event that an alternative subsystem fails.

DEPR:
As depicted in the exemplary embodiment, system 102 may include some level of redundancy to eliminate single points of failure. For example, each node 104-110 may be connected to each public network 112-114 by two network adapters (not shown): a service adapter providing the primary active connection between a node and network and a standby adapter which substitutes for the service adapter in the event that the service adapter fails. Thus, when a resource within system 102 becomes unavailable, alternative resources may be quickly substituted for the failed resource.

DEPR:
Those of ordinary skill in the art will appreciate that the hardware depicted in the exemplary embodiment of FIG. 1 may vary. For example, a system may include more or fewer nodes, additional clients, and/or other connections not shown. Additionally, system 102 in accordance with the present invention includes reliable communications and synchronizations among data processing systems 104-110, and an integrated cluster system management facility, described in further detail below.

DEPR:
The exemplary embodiment of FIG. 2A depicts nine data processing systems 202-218 organized as four resource groups 220-226. Within each resource group 220-226, typically only one data processing system manages a given application for that resource group at any given time. However, other data processing systems are designated to assume management of the application should the primary data processing system fail. A configuration object for each resource group 220-226 is replicated to each data processing system within the resource group. Each data processing system within the resource group is listed as an owner of the configuration object for the resource group. The configuration object contains cluster configuration and status information relevant to the resource group or resource, which includes: topology information such as data processing systems, networks, network interface cards (adapters), and network connectivity information; resource group information such as application packages for an application type of resource, shared disks for a shared disk type of resource, data processing system and disk connectivity information, service IP addresses for a service IP address types of resource, data processing systems where applications are installed and configured, management policies, management rules, and resource dependency relationships; and cluster system status information such as status of data processing systems, status of networks, status of network interface cards, status of shared disks, status of applications, and status of event processing. A configuration object may also contain rules for adding/modifying/deleting data processing systems, networks, network interface cards, shared disks, resource groups, and resources, as well as rules for evaluating resource dependency.

DEPR:
The complexity of managing a resource group having M data processing systems is M.sup.2, and since M is usually much smaller than the size N of a large cluster, significant performance improvements may be achieved both in replicating a configuration and status database and in access information in a database distributed among the M data processing systems. The response time for managing system events is significantly faster sing the complexity of cluster system management has been reduced by a factor of (M/N).sup.2. With the approach of the present invention, both the number of messages transmitted in a two-phase commit protocol to update a configuration and status database and the database access time are reduced significantly by involving only a subset of data processing systems within the cluster.

DEPR:
A separate, cluster configuration database may be implemented on top of the resource group configuration database. The cluster configuration database would be replicated to all data processing systems within the cluster and contain cluster configuration and status information regarding networks, data processing systems, cluster system events, etc.

DEPR:
The partitioning of the nine-node example depicted in FIGS. 2A-2H in accordance with the present invention will result in a seven different configuration databases. A simplified example of the configuration database managed by node group 228 would be:

DEPR:
A simplified example of the configuration database managed by node group 240 would be:

DEPR:
A simplified example of the configuration database managed by node group 236 would be:

DEPR:
As an example of recovery is such a partitioned system, suppose node 208 should fail. The recovery_status of node 208 is modified to `down` by the remaining group members of group 234, which includes nodes 202, 214, 210, and 212. The resulting configuration database for node group 234 is:

DEPR:
As an example of quorum condition within resource groups, supposed the entire nine-node cluster is restarted and initially only nodes 202 and 208 are up and running. The application ha_resource_group_220, which is managed by group 228, has reached quorum condition. Nodes 202 and 208 may determine between themselves which node should run ha_resource group_220. This approach allows ha_resource_group_220 to run without compromising data integrity even though the cluster as a whole does not have quorum--i.e. only 2 nodes are up among the total of nine nodes. The application ha_resource_group_226, on the other hand, which is managed by group 240, has one one node (node 208) within the group, and therefore does not have quorum condition.

DEPR:
The partial replication management approach of the present invention also handles catastrophes such as network partitions better than a centralized or fully replicated scheme. With partial replication of configuration and status information only among resource group owners, each resource group within a cluster may provide services if more than one half of the data processing systems within the corresponding owner list are online. Therefore, a cluster with partial replication of configuration information may continue to provide reliable services even if broken into small pieces, each much smaller than a quorum of all data processing systems in the cluster.

DEPR:
By partitioning the configuration database and allowing each sub-cluster of

servers to manage their configuration, a sub-cluster of servers may start providing services when it reaches "quorum," which may occur before the cluster as a whole reaches quorum. The "quorum" of resource group nodes which must be online need not necessarily be a majority of the node in the resource group, provided that at least one service may be reliably provided by the resource group. Furthermore, it may happen that the cluster may not be able to reach quorum if, for example, multiple failures occur. In such a case, sub-clusters may continue to provide their services as long as they have quorum. This is an advantage accompanying the partial replication method of the present invention, which associates quorum condition with each resource group while existing schemes associate quorum with the cluster as a whole.

DEPR:
With reference now to FIG. 3, a high level flowchart for a process of replicating configuration and status information within a cluster containing resource groups in accordance with a preferred embodiment of the present invention is depicted. The process begins at step 302, which illustrates a change in configuration or status data for a resource within the cluster system. The process then passes to step 304, which depicts a determination of whether the change is a "cluster-level" change, or a change which should be replicated throughout the cluster system. Some changes in configuration and status information--e.g., failure or reintegration of a node--should be replicated throughout the entire cluster system. For example, if a node is added to the cluster system, all pre-existing nodes, regardless of which resource groups contain the nodes, should be updated to reflect that addition. If the configuration and status information change is a cluster-level change, the process proceeds to step 306, which illustrates replicating the change throughout the cluster system

DEPR:
If the configuration and status information change is not a cluster-level change, the process proceeds instead to step 308, which depicts replicating the change among the node within the resource group affected by the change. Configuration and status information changes which affect only an application or the associated resource group need only be replicated throughout the resource group. A resource group manager, which may simply be the node within the resource group currently having the highest precedence, is utilized to insure proper replication of the configuration and status information change.

DEPR:
The process next passes to step 310, which illustrates a determination of whether a node within the resource group is shared with another resource group. If so, the process proceeds to step 312, which depicts replicating the configuration and status change to all nodes within the other resource group or groups. The node or nodes shared by the different resource groups are responsible for insuring proper replication. In this respect, interlocking resource groups within the cluster system are undesirable since it requires additional replication of configuration and status information. Further replication is not necessary, however, so that the change need not be replicated to resource groups within the cluster system which have no nodes in common with the resource group affected by the change.

DEPR:
Once the information is fully replicated among all nodes within the affected resource group or resource groups having at least one node in common with the affected resource group, or if the affected resource group does not include any nodes shared with another resource group, the process proceeds to step 314, which illustrates the process becoming idle until a subsequent configuration and status information change is detected.

DEPR:
Referring to FIG. 4, a high level flowchart for a process of handling node failure within a cluster system including resource groups in accordance with a preferred embodiment of the present invention is illustrated. The process begins at step 402, which depicts failure of a node within a resource group.

The process then passes to step 404, which illustrates a determination of whether a "quorum" of the resource group (or resource groups, if the failed node was shared) are available. As described above, the quorum need not be a majority, as long as sufficient resources are available within the resource group to reliably provide the service or services for which the resource group is defined.

DEPR:
If a quorum of nodes within the resource group is available, the process proceeds to step 406, which depicts continuing providing services utilizing available nodes. Some reallocation of resources may be necessary. The process then passes to step 408, which illustrates a determination of whether the failed node has been restored. If not, the process simply returns to step 408. If so, however, the process proceeds to step 410, which depicts reintegrating the node and reallocating resources as necessary.

DEPR:
Referring again to step 404, if a quorum of nodes is not available, the process proceeds instead to step 412, which illustrates suspending services from the affected resource group. The process then passes to step 414, which depicts a determination of whether the failed node has been restored. As described above, if the failed node has not yet been restored, the process simply returns to step 414. Once the failed node is restored, however, the process proceeds to step 416, which illustrates reintegrating the node and resuming services from the resource group affected. From either of steps 410 or 416, the process passes to step 418, which depicts the process becoming idle until another node failure occurs.

DEPL:
A simplified example of the configuration database managed by node group 230 would be:

DEPL:
A simplified example of the configuration database managed by node group 232 would be:

DEPL:
A simplified example of the configuration database managed by node group 234 would be:

DEPL:
And finally, a simplified example of the configuration database managed by node group 238 would be:

DEPL:
The application ha_resource_group_226, which was running on node 208 must be restarted on some other node. This application is managed by resource group 240 and therefore may be restarted on either node 210 or node 212. If node 210 is selected by the two remaining nodes in resource group 240 to run ha_resource_group_226, the resulting configuration database for node group 240 would be:

CLPV:
at least one network connecting the data processing systems in the cluster system;

CLPV:
a network connection permitting the data processing system to be connected to a cluster at system segregated into a plurality of resource groups;

CLPV:
instructions embodied within said computer usable medium, for segregating data processing systems in a network into at least one resource group, each resource group including at least two data processing systems and related resources for providing a respective computing service;